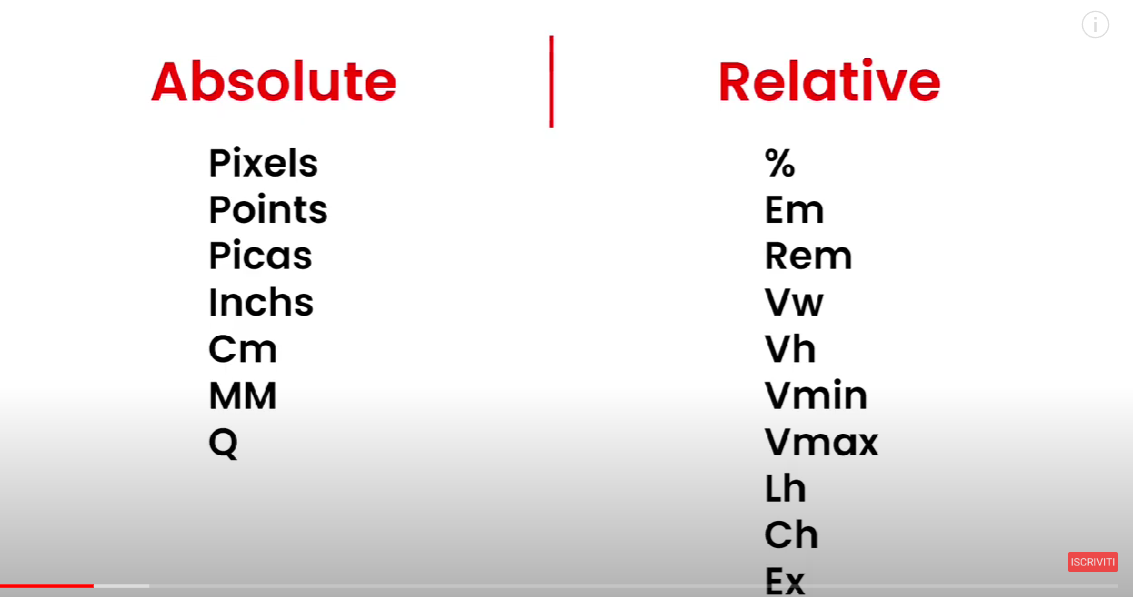
**UNITA DI MISURA**

**Le unità di misura dei nostri elementi html si suddividono in due categorie:**

* **Unità di misura assolute**
* **Unità di misura relative**

**Vediamo qui sotto una slide che riporta le unità di misura assolute e relative. Ovviamente non vedremo tutte queste unità di misura, ma solo quelle più utilizzate.**

****

**UNITA DI MISURA ASSOLUTE**

**Le unità di misura assolute sono quelle base. Qui sotto è riportata una tabella con le unità di misura assolute.**

****

**In particolare, si ha che 1px è pari a:**

* **0.75 Pt**
* **0.0264583333 cm**
* **0.2645833333 mm**
* **1 pixel = 1 inch / 96 = 0.010417 inch**

**Si noti che il valore in pixel (o in cm, mm, Pt, inch) associato ad una proprietà non cambia mai.**

**UNITA DI MISURA RELATIVE**

**Le unità di misura relative sono quelle più utilizzate in CSS. Queste unità di misura indicano la grandezza di un elemento rispetto alla grandezza di un altro.**

**Supponiamo di avere due div, e che un div sia contenuto nell’altro. Chiamiamo A il div contenitore e B il div contenuto.**

**Supponiamo che la larghezza del div contenuto sia il 40% della larghezza del div contenitore. Se la larghezza del div contenitore si riduce, si ridurrà anche la larghezza del div contenuto.**

**Vediamo quindi quali unità di misura relative andremo a studiare.**

* **%**
* **em**
* **rem**
* **vw**
* **vh**

**Percentuale %**

**Dati due div, A e B, con B contenuto in A, ha senso applicare la percentuale sulla larghezza o sull’altezza del div B.**

**Utilizziamo il seguente esempio per capire il funzionamento della percentuale su un div contenuto in un altro div**

**HTML**

<body>

    <div class="parent">

        <div class="figlio">

        </div>

    </div>

</body>

**CSS**

.parent{

    width:700px;

    height:500px;

    border:2px solid blue;

}

.figlio{

    width:70%;

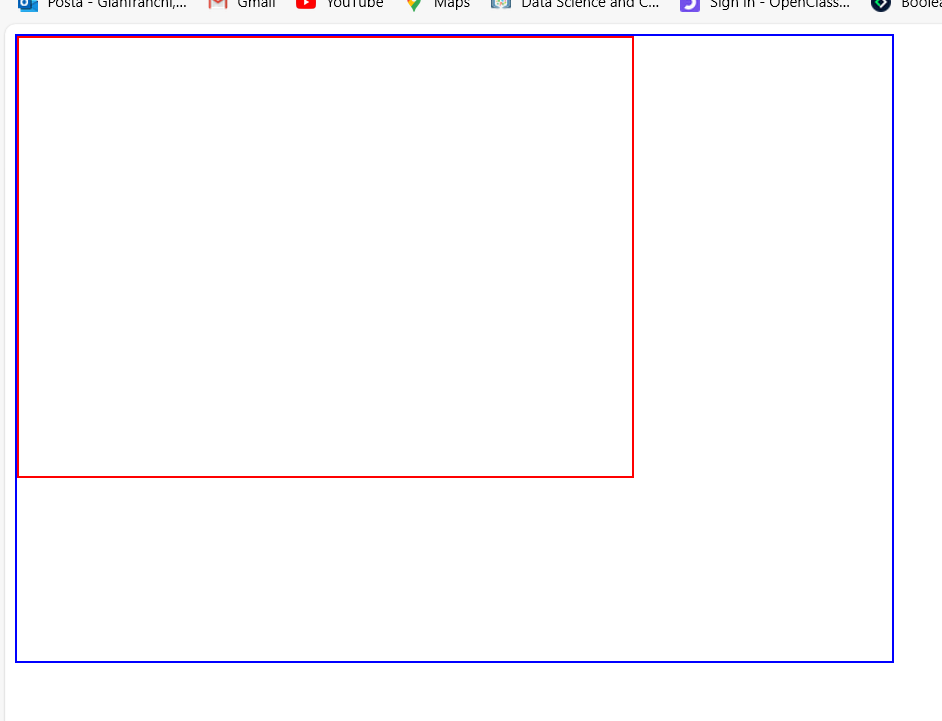
    height:70%;

    border:2px solid red;

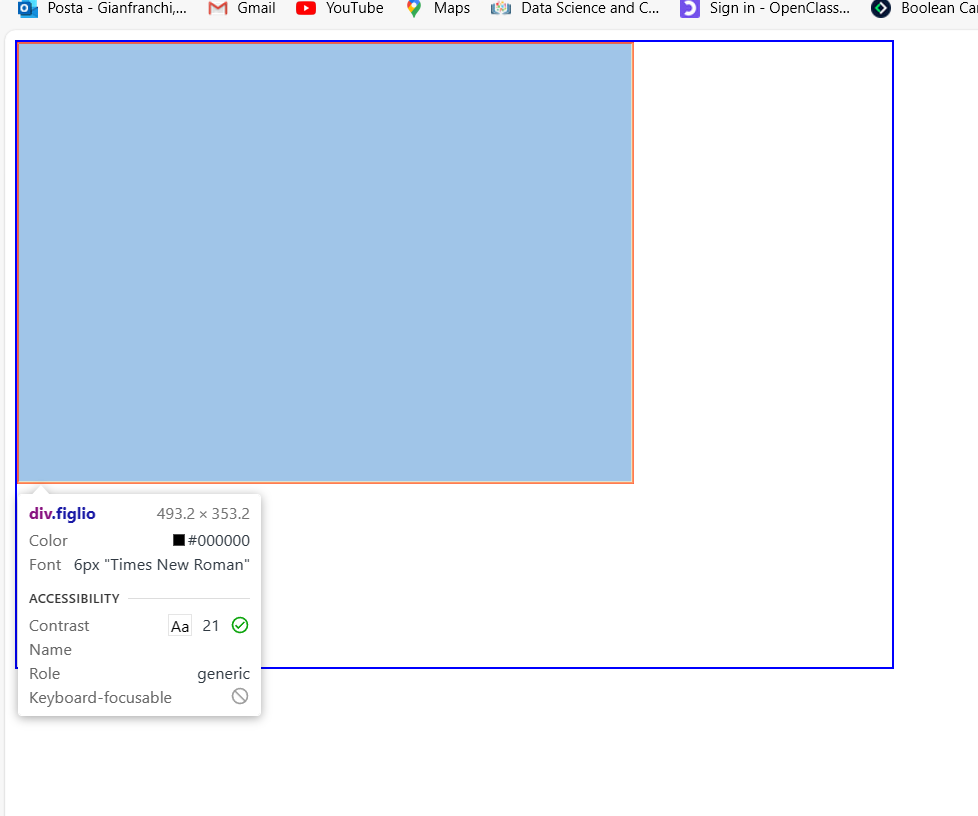
    font-size:28%;

}

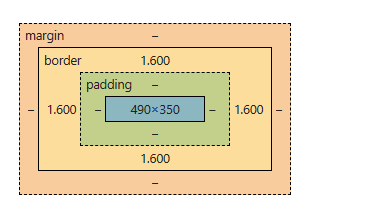
**BROWSER**

****

**Apriamo l’inspect sul div figlio:**

****

**E incolliamo qui il suo box-model:**

****

**Si ricordi che la larghezza totale di un elemento HTML è data dalla seguente formula:**

**WIDTH\_TOTALE = WIDTH + MARGIN + PADDING.**

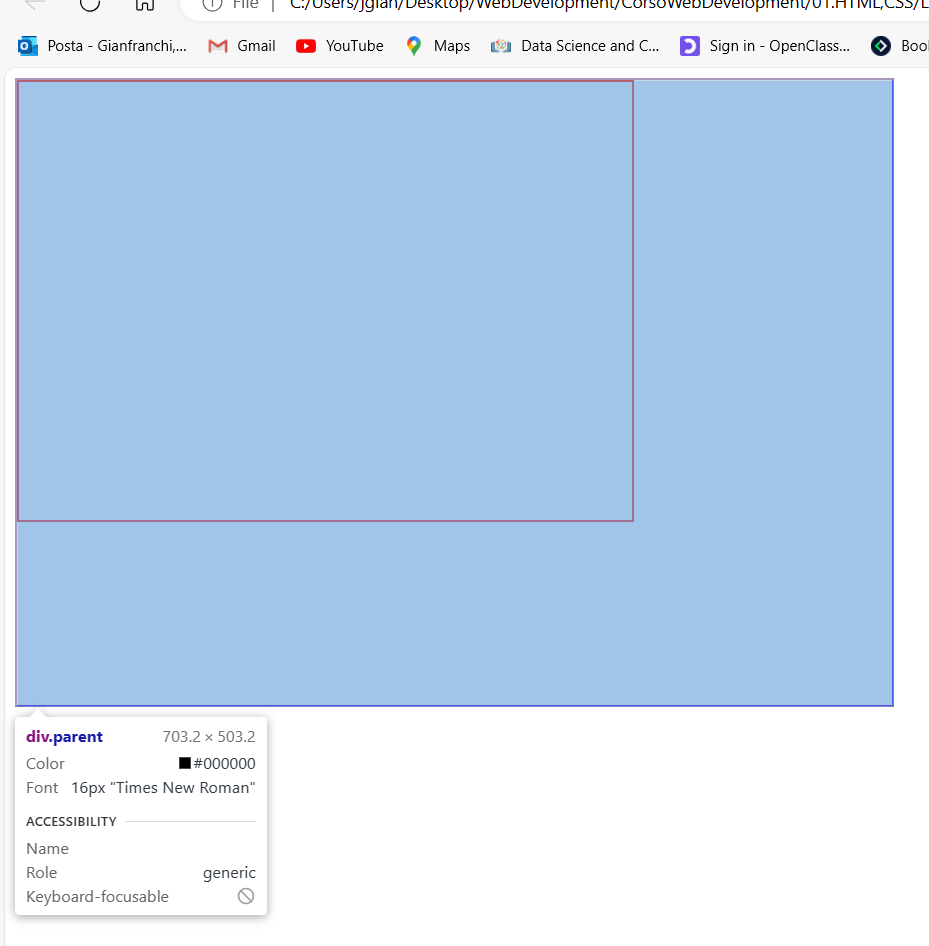
**Dove:**

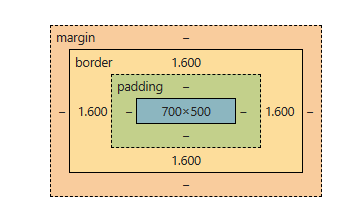
**MARGIN = MARGIN\_LEFT + MARGIN\_RIGHT**

**PADDING = PADDING\_LEFT + PADDING\_RIGHT**

**Poiché nel nostro caso MARGIN = 0 E PADDING = 0, si avrà che WIDTH\_TOTALE = WIDTH.**

**E ora apriamo l’inspect sul div parent e incolliamo sotto il box-model**

****

****

**Osservando il nostro codice, abbiamo che la width del div figlio è pari al 70% della width del div parent, ossia il 70% di 700 che è pari a 490px. Osservando i box model notiamo che le misure combaciano.**

**Aggiungiamo ora un margin ed un padding al nostro div parent:**

**CSS**

.parent{

    width:700px;

    height:500px;

    border:2px solid blue;

    padding:20px;

    margin:20px;

}

.figlio{

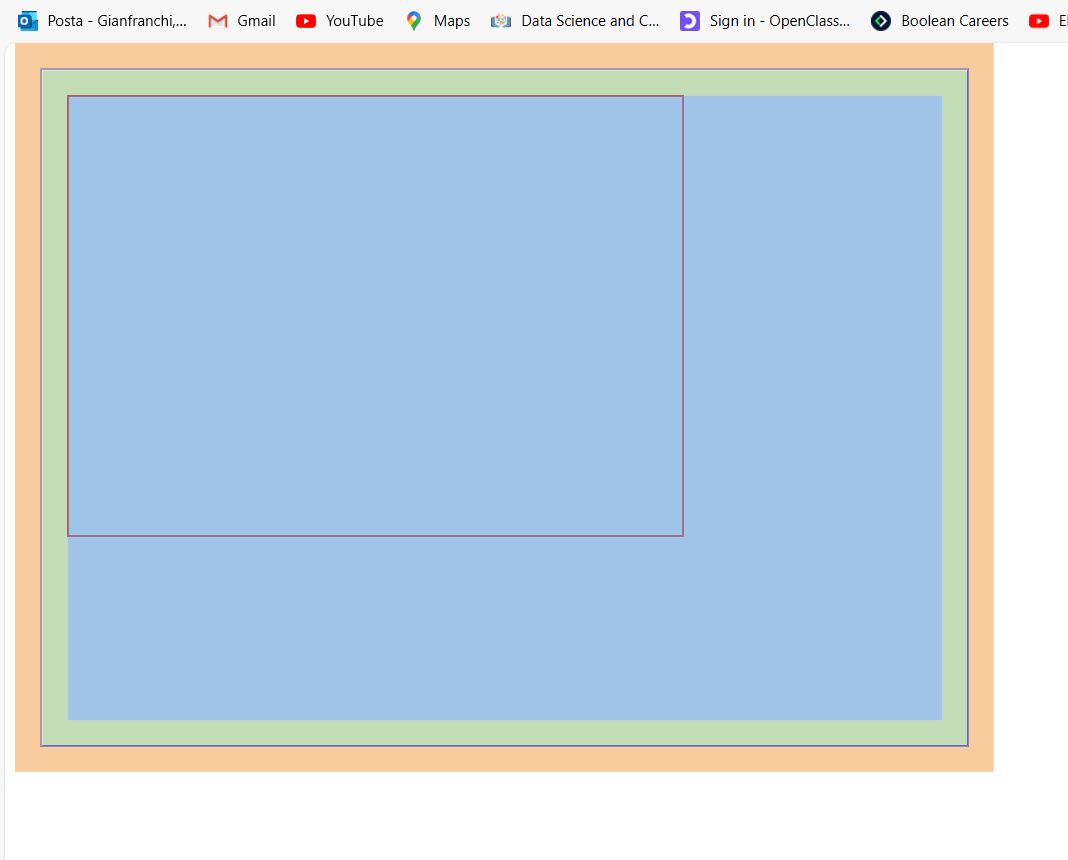
    width:70%;

    height:70%;

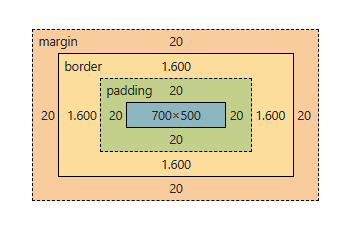
    border:2px solid red;

    font-size:28%;

}

**BROWSER (Relativo al div parent)  
**

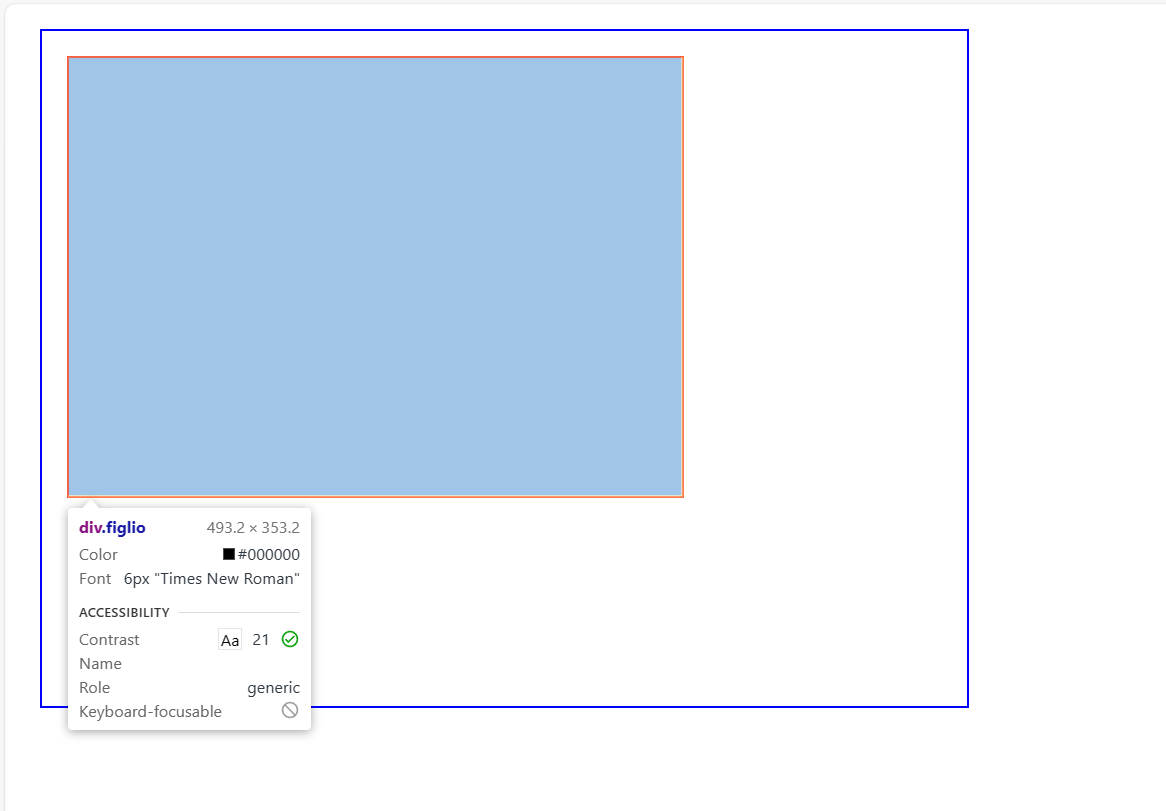
**E il box-model del div parent sarà:**

****

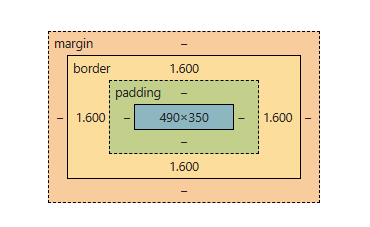
**Dal box-model si evince che la, dato che la larghezza (width) del div parent è sempre 700px, la largezza del div figlio sarà sempre il 70% di 700, ossia 490.**

**Il box model del div figlio dimostrerà quanto appena detto**

**BROWSER (Relativo al div figlio)**

****

**BOX MODEL (Relativo al div figlio)**

****

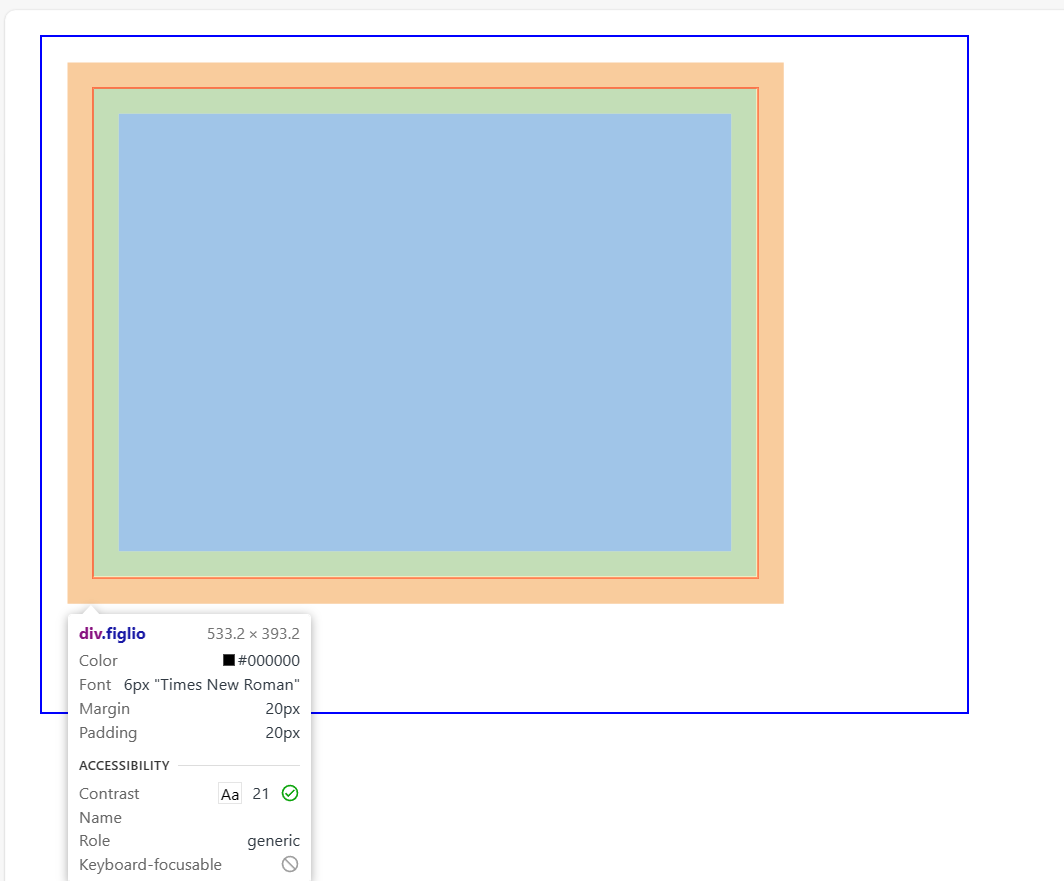
**In sintesi, margin e padding del div parent non vengono presi in considerazione.**

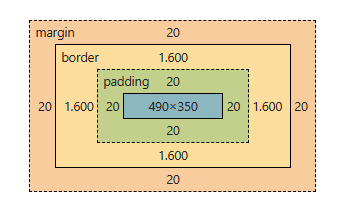
**Poiché la width totale di un elemento HTML è data da WIDTH + MARGIN + PADDING, si nota la width espressa in percentuale di un elemento interno ad un altro elemento farà riferimento alla width del div parent, e non alla sua width totale.**

Vediamo ora cosa succede se inseriamo un padding al nostro div figlio.

**In questo caso, la width totale del div figlio non coinciderà più con la sua width, ma sarà pari a WIDTH + PADDING.**

**Apriamo quindi l’inspect su di esso e riportiamo qui sotto anche il box-model**

****

****

**La width è quindi, come al solito, pari a 490px (70% di 700)🡺 anche in questo caso ci si riferisce alla width, e non alla width totale.**

**Vediamo se anche la width totale del div figlio è pari al 70% della width totale del parent:**

**WIDTH\_TOTALE\_FIGLIO = 490+40+40 = 490 + 80 = 570px**

**WIDTH\_TOTALE\_PARENT = 700 + 40+40 = 700 + 80 = 780px**

**Il 70% di 780 è pari a 546px 🡺 WIDTH\_TOTALE\_FIGLIO è diversa dal 70% della WIDTH\_TOTALE\_PARENT!!!**

**Unità di misura em**

**Il suo nome proviene dalla tipografia, in cui, viene utilizzata per identificare la dimensione della lettera maiuscola M di un carattere. Tale unità di misura calcola la dimensione del testo in base ad una misura fissa (e quindi non relativa) di base.**

**Nel nostro caso, em indica una grandezza in corrispondenza del font-size dell’elemento contenitore, oppure del div stesso.**

**Se il div contenitore è il body, avremo che 1em corrisponde a 16px.**

**Vediamo un esempio**

**HTML**

<body>

<div>

    Buongiorno a tutti!

    <span class="figlio"></span>

</div>

</body>

**CSS**

.figlio{

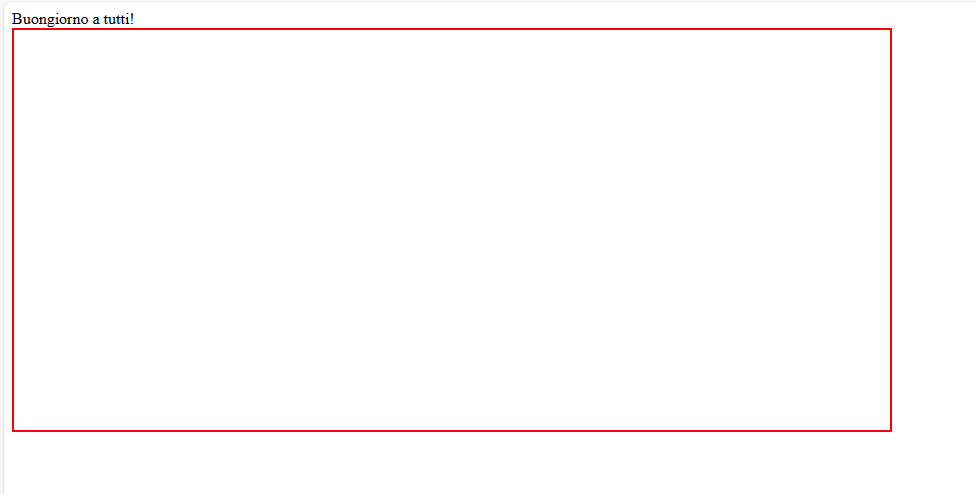
    width:70%;

    height:400px;

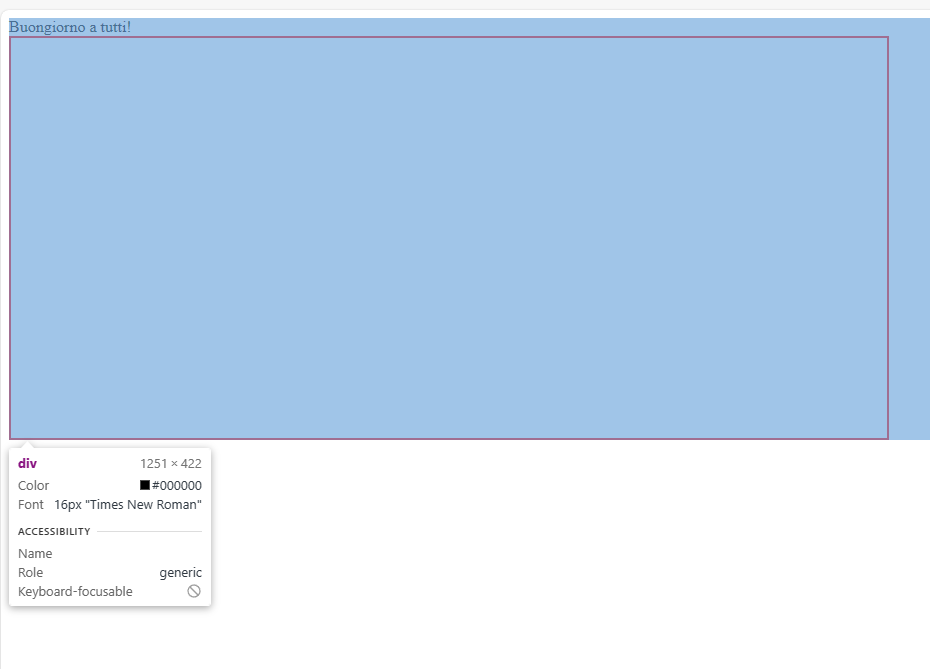
    border:2px solid red;

}

**BROWSER**

****

**Per prima cosa vediamo quanto è il font size del div contenitore:**

****

**Il font-size del div contenitore è pari a 16px, e quindi è uguale al font-size del body. Questo perché il div contenitore non ha font-size specificato 🡺 Eredita il font-size del body.**

**Supponiamo di volerlo cambiare. In questo modo, poiché il font-size viene assegnato, non erediterà più il font-size del body. Supponiamo di volerlo porre pari a 40px:**

**HTML**

<body>

<div>

    Buongiorno a tutti!

    <div class="figlio"></div>

</div>

</body>

**CSS**

div{

    font-size:40px;

}

.figlio{

    width:70%;

    height:400px;

    border:2px solid red;

}

**BROWSER**

****

**Si noti che, finora, abbiamo imposto che la larghezza del div contenuto fosse pari al 70% della larghezza del div container. Supponiamo ora di volerla porre pari al doppio della larghezza del font-size del div parent. Come fare? Semplice!**

**Si pone la larghezza del div figlio pari a 2em, ossia pari al doppio del font-size del div parent!**

**HTML**

<body>

<div>

    Buongiorno a tutti!

    <div class="figlio"></div>

</div>

</body>

**CSS**

div{

    font-size:40px;

}

.figlio{

    width:2em;

    height:400px;

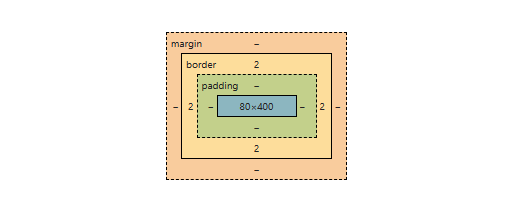
    border:2px solid red;

}

**BROWSER**

****

**Perché la larghezza del div contenuto è 84 invece di 80? Perché come al solito noi vediamo a schermo la larghezza complessiva. Se osserviamo il box-model, noteremo che la larghezza del div contenuto è effettivamente pari 80px!**

****

**CVD.**

**Ovviamente, l’esempio visto ora ha poco senso dal punto di vista pratico. Nel corso delle nostre lezioni e dei nostri progetti vedremo come utilizzarlo nella pratica.**

**In linea generale, comunque, l’unità di misura em si utilizza per calcolare il font-size di un tag rispetto al font-size del tag contenitore.**

**Vediamo qui un esempio banale.**

**HTML**

<body>

    <div id="container">

        <p>HOLAAAA</p>

    </div>

</body>

**CSS**

#container{

    font-size:40px;

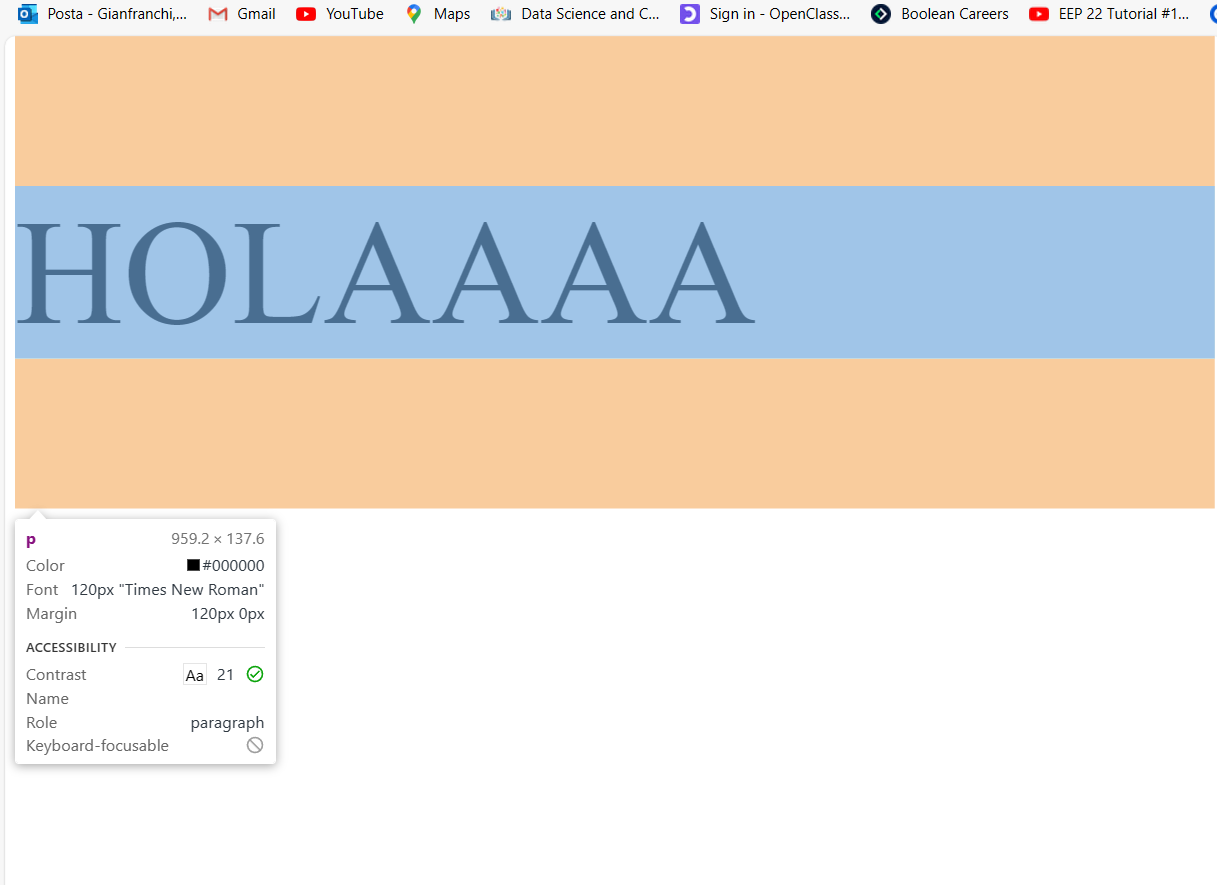
}

p{

    font-size:3em;

}

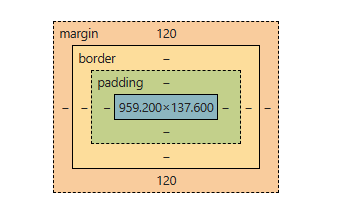
**BROWSER**

****

**Tutto perfetto. Il font-size del testo contenuto nella p contenuta nel div è pari a 120px, ossia 3\*40.**

**Utilizziamo ora gli em per calcolare il margin della nostra p.**

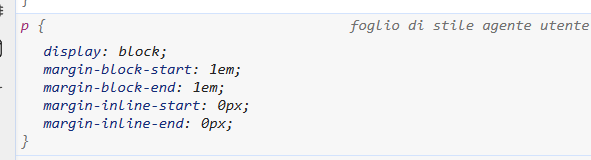
**Osserviamo un attimo il box-model relativo al nostro paragrafo p:**

****

**Margin-top = 120px e margin-bottom:120px.**

**Perché?**

**Osserviamo I valori di default della nostra p:**

****

**margin-block-start corrisponde a margin-top, mentre margin-top-end corrisponde a margin-bottom.**

**Si noti che essi corrispondono a 1em.**

**In base a quello che abbiamo studiato fino ad oggi, 1em dovrebbe indicare che il margin-top della nostra p dovrebbe essere 1 x font-size del div contenitore, ossia 1 x 40 = 40px.**

**Tuttavia, il nostro box-model mostra un altro valore, ossia 120px.**

**Perché?**

**Semplice: l’unità di misura em associata a margin, padding e border non usa come metro di paragone il font-size del tag contenitore, bensì usa il font-size del tag in cui ci si trova!!!**

**La nostra p presenta font-size=3em 🡺 3\*40 🡺 120px.**

**Quindi, il margin-top e il margin-bottom è pari a 120px!!!**

**Provando a cambiare manualmente il font-size della nostra p e modificando i suoi margini, i suoi padding e i suoi border:**

**CSS**

#container{

    border:1px solid blue;

    width:50%;

    font-size:40px;

}

p{

    border-top: 0.15em solid red;

    border-bottom: 0.1em solid red;

    border-left: 0.16em solid red;

    border-right:0.2em solid red;

    margin-top:2em;

    margin-bottom:1.4em;

    margin-left:2.2em;

    margin-right:0.68em;

    padding-top:0.30em;

    padding-bottom:0.22em;

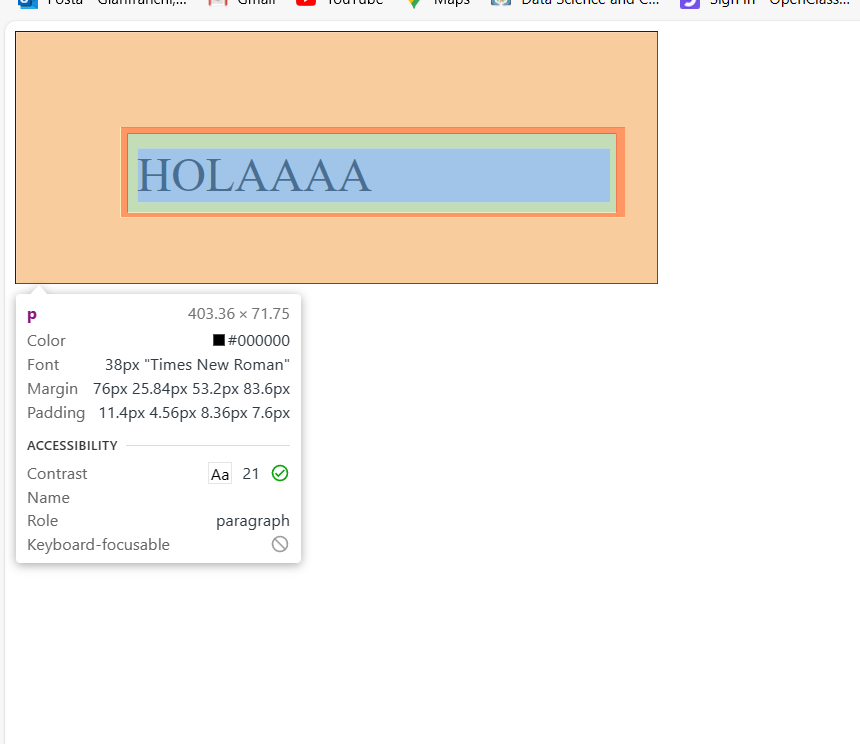
    padding-left:0.2em;

    padding-right:0.12em;

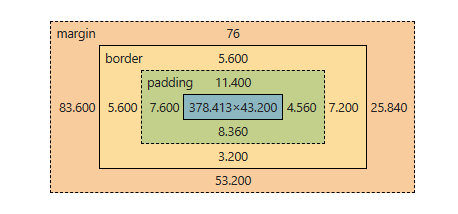
    font-size:38px;

}

**BROWSER**

****

**Vediamo il box-model**

****

**Studiamo quindi i nostri margini e i nostri padding:**

**MARGINI**

**margin-top:76px**

**margin-bottom:53.200px**

**margin-left:83.6px**

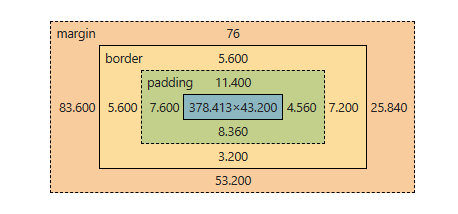
**margin-right:25.84px**

**Il nostro paragrafo ha font-size pari a 38px.**

**Andiamo quindi a dimostrare il perché i nostri margini hanno le dimensioni scritte qua sopra.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Margine** | **Font-size** | **Dimensione in em** | **Calcolo** | **Risultato** |
| **margin-top** | **38px** | **2em** | **2\*38** | **76px** |
| **marign-bottom** | **38px** | **1.4em** | **1.4 \* 38** | **53.2px** |
| **margin-left** | **38px** | **2.2em** | **2.2 \* 38** | **83.6px** |
| **margin-right** | **38px** | **0.68em** | **0.68\*38** | **25.84px** |

**Per comodità riportiamo qui sotto il box-model:**

****

**Perfect!!**

**PADDING**

**padding-top:11.4px**

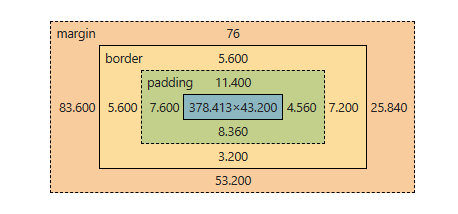
**padding-bottom:8.36px**

**padding-left:7.6px**

**padding-right:4.56px**

**Il nostro paragrafo ha font-size pari a 38px.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Padding** | **Font-size** | **Dimensione in em** | **Calcolo** | **Risultato** |
| **padding-top** | **38px** | **0.3 em** | **0.3\*38** | **11.4px** |
| **padding-bottom** | **38px** | **0.22 em** | **0.22 \* 38** | **8.36** |
| **Padding-left** | **38px** | **0.2em** | **0.2 \* 38** | **7.6px** |
| **Padding-right** | **38px** | **0.12em** | **0.12\*38** | **4.56px** |

****

**Perfect!!**

**Unità di misura rem**

**L’unità di misura rem è molto simile all’unità i misura em, ma invece di utilizzare come metro di paragone il font-size dell’elemento parent, in questo caso si utilizza il font-size del body. Vediamo subito un semplice esempio.**

**HTML**

    <body>

        <p>CIAO A TUTTIIII</p>

    </body>

**CSS**

\*{

    margin: 0;

    padding:0;

}

p{

    font-size:3rem;

}

**BROWSER**

****

**Si ricordi che il font-size del body è pari a 16px di default 🡺Il font size del testo contenuto nel nostro paragrafo dovrebbe essere pari a 16\*3px, ossia a 48px.**

**Vediamo se è vero (rullo di tamburi…)**

****

**Perfetto!!!**

**E se provassimo a cambiare il font-size del body?**

**CSS**

\*{

    margin: 0;

    padding:0;

}

body{

    font-size:25px;

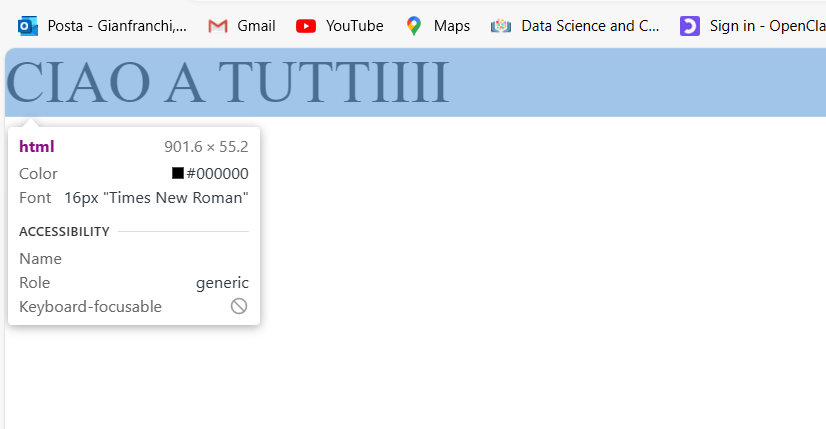
}

p{

    font-size:3rem;

}

**BROWSER**

****

**Da quanto si evince dallo screen precedente, non è possibile modificare il font-size del body.**

**Per concludere l’argomento rem proviamo a vedere se, anche nel caso di div innestati, il metro di paragone del testo contenuto in div innestato è sempre il font-size del body, ossia 16px.**

**HTML**

   <div class="div-container">

        <span>CIAO</span>

            <div class="div1">

                <span>HELLO</span>

                    <div class="div2">

                            <span>HOLA</span>

                    </div>

            </div>

   </div>

**CSS**

\*{

    margin: 0;

    padding:0;

}

body{

    font-size:25px;

}

.div-container{

    font-size:20px;

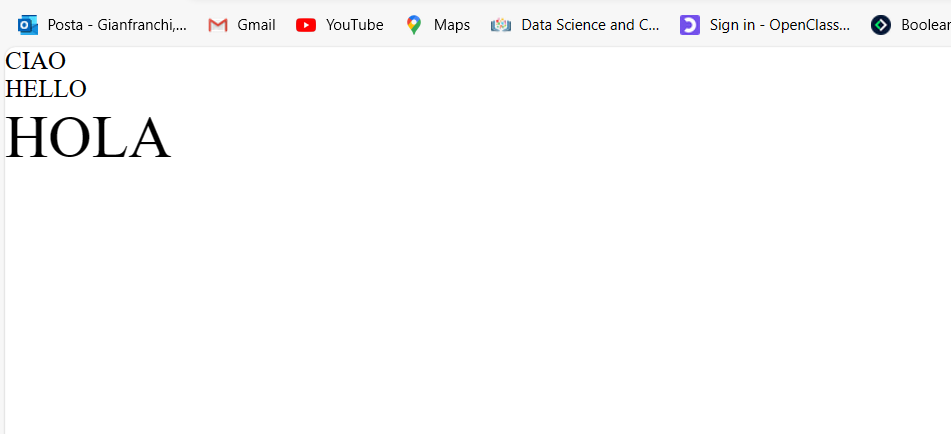
}

.div2{

    font-size:3rem;

}

**BROWSER**

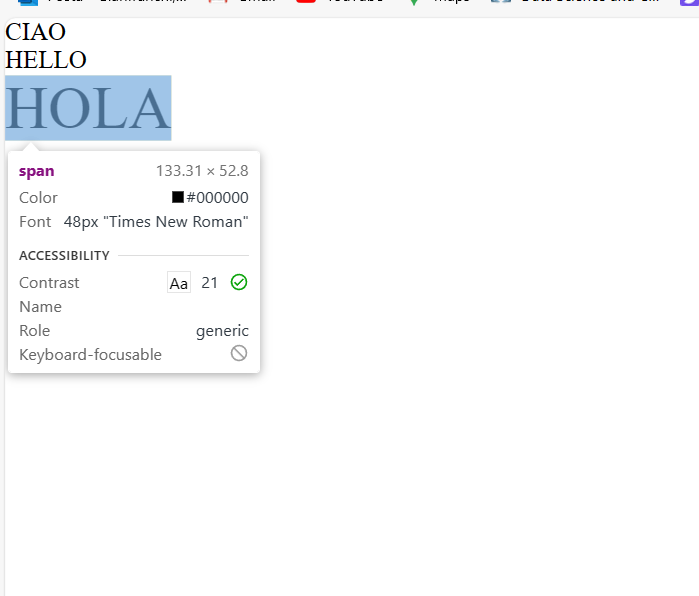
****

**Descriviamo un po’ cosa è successo.**

**Il div più esterno ha font-size pari a 20px 🡺 Tutti gli span interni, per ereditarietà, avranno font-size pari a 20px.**

**L’ultimo div, ossia quello più interno avente classe div2, è un po diverso. Esso ha font-size pari a 3em 🡺 Il test contenuto nello span contenuto in div2 avrà font-size uguale a 3 volte in font-size del body, ossia 3\*16 = 48px.**

**Sarà vero?**

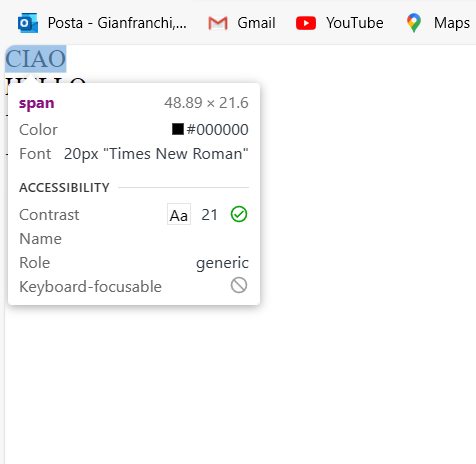
****

**Yes!**

**Ovviamente in questo esempio sono entrati in gioco il concetto di ereditarietà e di priorità degli elementi HTML. Infatti:**

* **Il div più esterno, ossia il div avente classe .div-parent, ha font-size:20px.**
* **Tale dimensione del font-size verrà ereditata dal testo CIAO contenuto nello span presente in .div-parent. Tale testo sarà quindi pari a 20px.**
* **Lo screen qui sotto mostrerà che quanto detto finora è vero:**

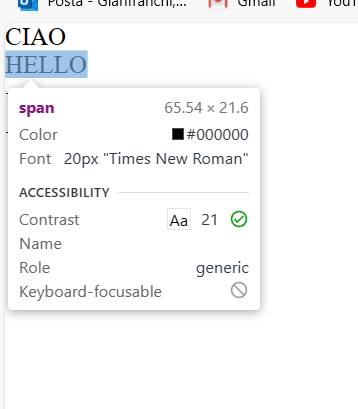
**BROWSER**

****

**Perfetto!**

* **Ora andiamo a vedere il testo HELLO, ossia il testo contenuto nello span contenuto nel div avente classe .div1.**
* **Il primo div interno .div1 eredita il font-size del div-parent, e questo si ripercuoterà nel testo contenuto nello span interno a .div1. Quindi, il testo CIAO contenuto nello span contenuto in .div1 erediterà il font-size di .div-parent, e sarà pari a 20px. Vedremo dallo screen qui sotto se quanto detto è vero.**

**BROWSER**

****

**Perfect!**

* **Infine, vediamo il div .div2 e lo span al suo interno.**
* **Da quanto si evince dal codice HTML, abbiamo applicato la proprietà font-size a .div2. Poiché la proprietà font-size si trova sia nel div più esterno (.div-parent) sia nel div più interno (.div2) essa sarà sovrascritta.**
* **In sintesi, il div .div-parent ha font-size:20px, mentre il div .div2 ha proprietà font-size:3em. Quindi, .div2 per prima cosa eredita la proprietà font-size:20px dal div più esterno .div-parent. Dopodiché la sovrascrive con font-size:3em**
* **3em significa che il font-size del div .div2 sarà pari a 3 volte il font-size del body, ossia 3\*16 = 48px.**

**BROWSER**

****

**CVD**

**Infine, proviamo a vedere cosa succede se applicassimo al div .div2 la misura 3em, invece che la misura 3rem.**

**In base a ciò che sappiamo, la dimensione del font-size del testo contenuto nello span contenuto nel div .div2 dovrà essere pari a 3 volte la dimensione del font-size del div contenitore, ossia del div .div1. Ma il div .div1 eredita il font-size dal div .div-parent, e tale font-size è pari a 20px.**

**Quindi, il font-size del div .div2 è pari a 3\*20 = 60px. Vediamo se è vero**

**BROWSER**

****

**Perfect!!**

**Prima di passare alla prossima unità di misura, ricordiamo brevemente alcuni punti.**

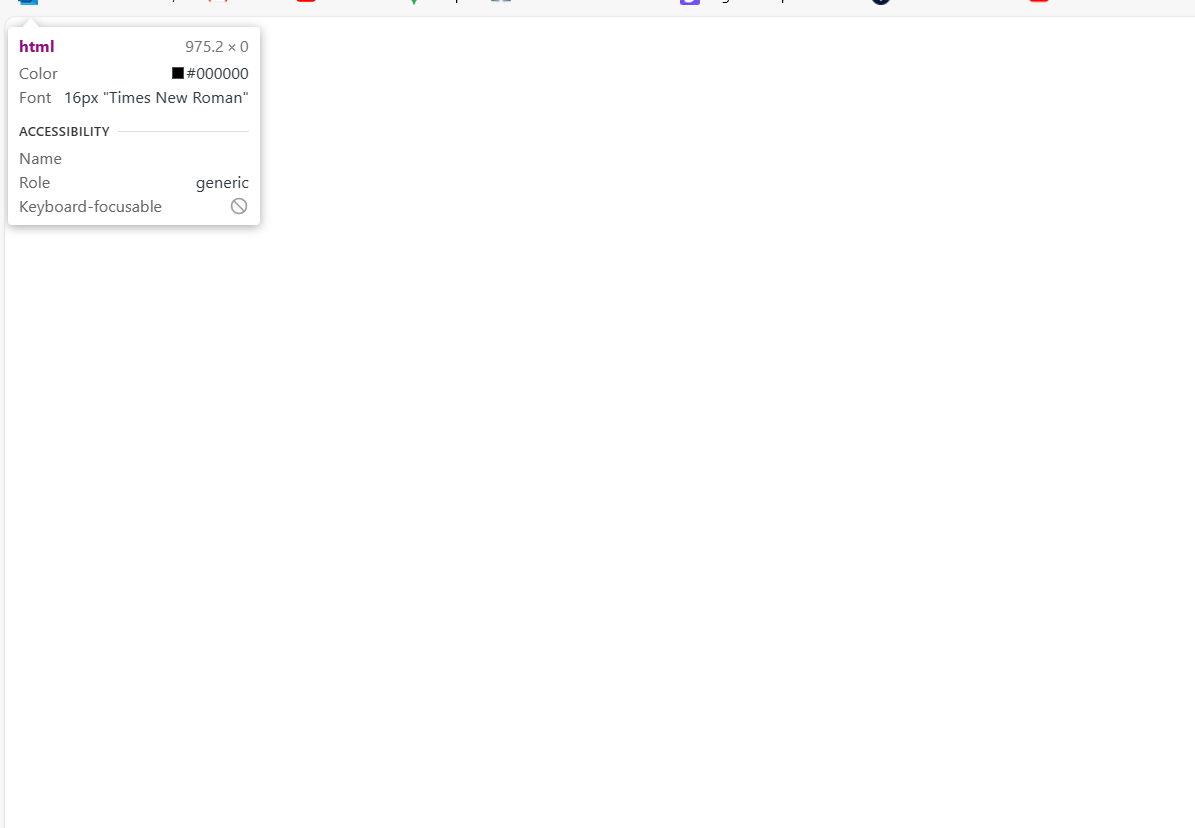
* **Il div più esterno, ossia nel nostro caso il div .div-parent, eredita inizialmente il font-size del body, ossia 16px. Dato che abbiamo posto il font-size del nostro div .div-parent pari a 20px, avremo un conflitto: Il nostro div .div-parent per prima cosa eredita il font-size dal body, pari a 16px, e successivamente lo sovrascrive con font-size:20px.**
* **Abbiamo applicato finora ai nostri em e rem un moltiplicatore intero. Ovviamente, possiamo avere anche moltiplicatori decimali, ad esempio 1.5rem, 2.3rem, e così via. I moltiplicatori, tuttavia, NON possono essere numeri negativi.**

**Unità di misura relative al viewport.**

**Per studiare le prossime unità di misura, dobbiamo introdurre il concetto di ViewPort.**

**Il ViewPort è la porzione di browser visibile dall’utente. Essa varia in base al device sul quale si vuole mostrare la pagina web, ad esempio pc, tablet, smartphone etc. In altre parole, il viewport è il nostro body**

**Vediamo un esempio di viewport.**

****

**Nel nostro esempio, il viewport ha larghezza 975.2px ed altezza pari a 0px. Perché la nostra altezza è 0px? Perché ricordiamo che essa, a meno che non venga specificata manualmente, prende la somma delle altezze dei contenuti. Poiché nel nostro esempio non abbiamo contenuti, l’altezza del viewport è pari a 0px.**

**Se aggiungessimo un elemento, vedremo che l’altezza del viewport cambierà.**

****

**Poiché il font-size di un h1 è pari a 2em, e poiché nel nostro caso il div contenitore è il body, si avrà che il font-size del nostro h1 sarà pari a 32px🡺 La sua altezza sarà 37.6px 🡺 Poiché h1 è l’unico elemento presente nel nostro browser, anche l’altezza del nostro viewport sarà uguale a 37.6px.**

**Siamo ora pronti a studiare le unità di misura relative al viewport.**

**In prima battuta studieremo le unità di misura vw (viewport width) e vh (viewport height)**

**Unità di misura vw**

**Ripassiamo quanto visto finora:**

* **Unità di misura %: Misura la width di un div in relazione alla width del div contenitore**
* **Unità di misura em: misura il font-size del testo contenuto in un tag in relazione al font-size del testo contenuto nel tag parent**
* **Unità di misura rem: misura il font-size del testo contenuto in un tag in relazione al font-size del viewport (16px)**

**Vedremo ora l’unità di misura vw: misura la width di un tag in relazione alla width del viewport.**

**Riprendiamo in considerazione l’esempio visto in precedenza, apportando alcune modifiche:**

**HTML**

    <div class="div-container">

        <span>CIAO</span>

            <div class="div1">

                <span>HELLO</span>

                    <div class="div2">

                            <span>HOLA</span>

                    </div>

            </div>

   </div>

**CSS**

\*{

    margin: 0;

    padding:0;

}

.div-container{

    border:1px solid red;

    width:70%;

    font-size:2em;

}

.div1{

    border:1px solid green;

    font-size:2em;

}

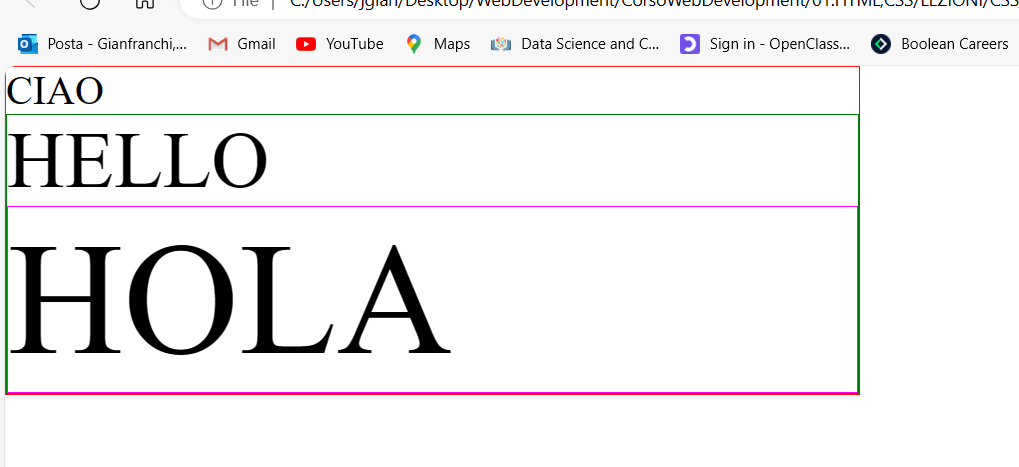
.div2{

    border:1px solid fuchsia;

    font-size:2em;

}

**BROWSER**

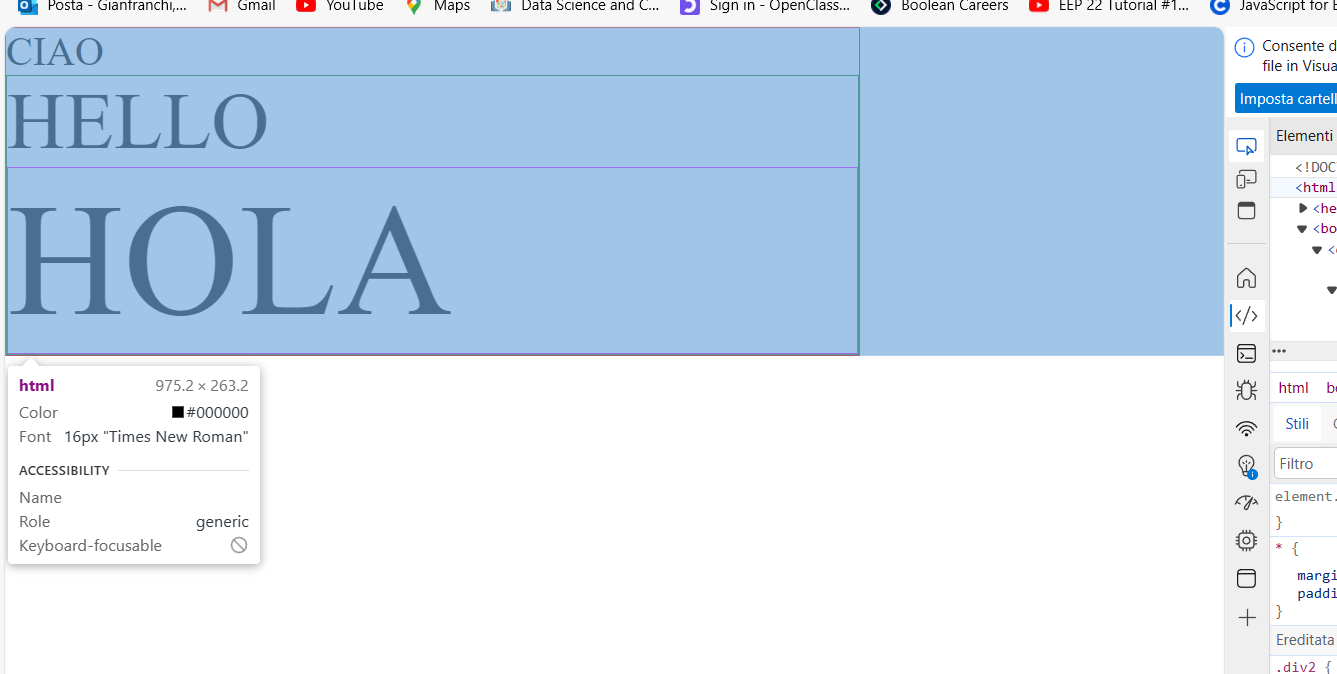
****

**I font-size del nostro esempio sono semplici da spiegare**

* **Il font-size del nostro div .div-container è pari a 2em, ossia 2 volte il font-size del nostro body 🡺 32px.**
* **Il font-size del div .div1 è pari a 2em, ossia a 2 volte il font size del div contenitore .div-container 🡺 2.32=64px.**
* **Nell’ultimo caso avremo quindi 2\*64=128px.**

**Le width dei nostri elementi sono anch’esse semplici da spiegare.**

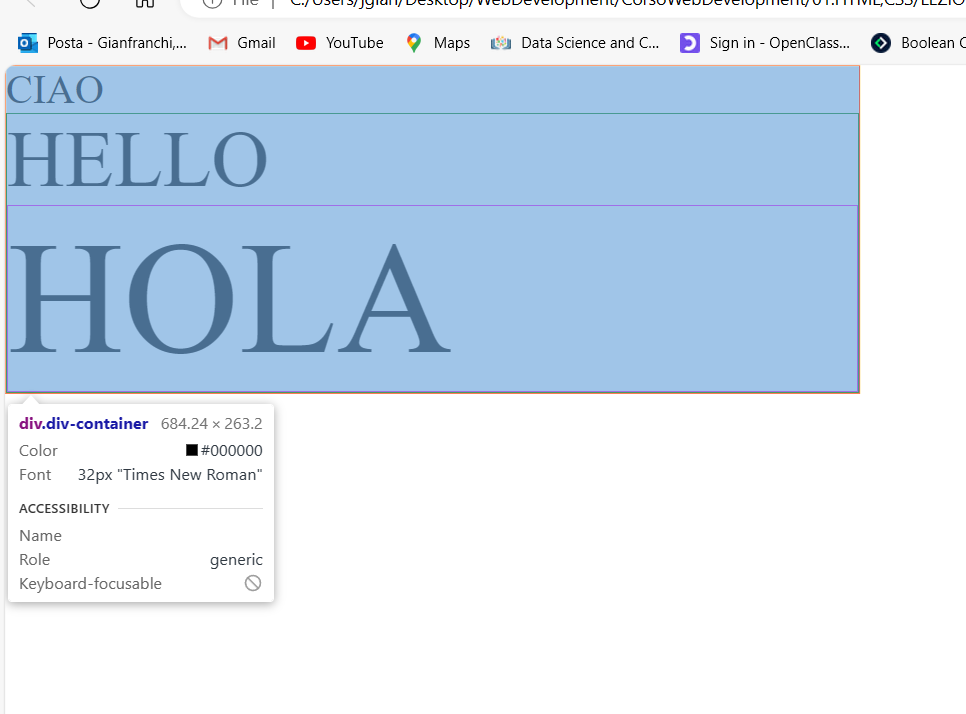
**Per prima cosa vediamo quant’è la width del nostro body/viewport:**

****

**Essa è quindi pari a 975.2px.**

**Il div .div-container ha width pari al 70% della width del suo contenitore, ossia del body 🡺 la width di div-container è pari a 70% di 975.2 = 682.64px.**

**Vediamo se è vero:**

****

**Il risultato è leggermente diverso per via della presenza dei bordi.**

**Si ricordi che, all’interno del nostro div .div-container abbiamo un div .div1 e, al suo interno, abbiamo un altro div .div2.**

**Poiché non è specificata la larghezza ne in div .div1 ne in div .div2, avremo che div1 erediterà la width dal div contenitore, ossia da .div-container, (quindi sarà pari a 684.42px come quella del container), e div2 la erediterà da div1, e anch’essa sarà quindi pari 684.42px (Come al solito le piccole differenze sono dovute dalla presenza dei bordi).**

**Vediamo ora il funzionamento di vw.**

**Se volessimo far si che il div .div1 abbia una larghezza non in base alla larghezza al div contenitore, ma in base alla larghezza del body, non si utilizza la percentuale % oppure rem, ma l’unità di misura vw.**

**Nel nostro esempio:  
  
CSS**

\*{

    margin: 0;

    padding:0;

}

.div-container{

    border:1px solid red;

    width:70%;

    font-size:2em;

}

.div1{

    border:1px solid green;

    width:80vw;

}

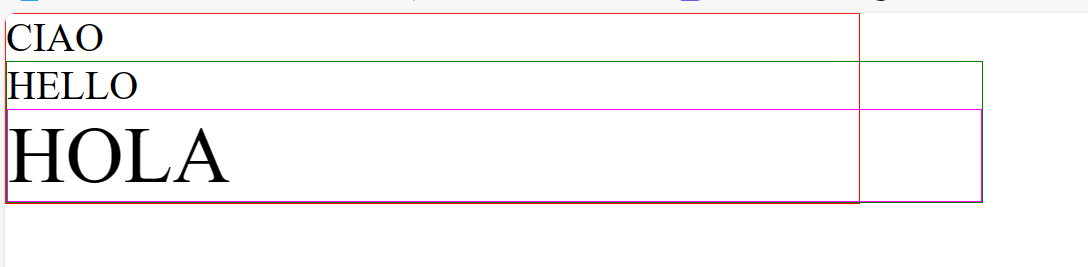
.div2{

    border:1px solid fuchsia;

    font-size:2em;

}

**BROWSER**

****

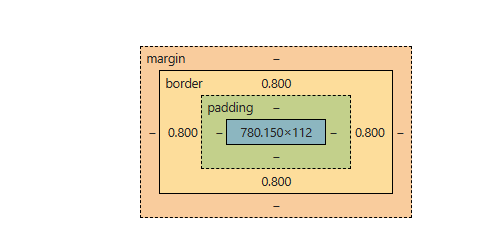
**Perfetto!**

**Il nostro div .div1 avrà quindi width pari all’80% della width del body, e il nostro div .div2 erediterà tale grandezza.**

**Ricordiamo che la larghezza del body è pari a 975.2px**

**Infatti, l’80% di 975.2 = 780.16**

**Che ci dice il box-model?**

****

**Perfetto!! La width è pari a 780.16**

**Rivediamo lo stesso argomento utilizzando misure diverse per i nostri div innestati:**

**HTML**

    <div class="div-container">

        <span>CIAO</span>

            <div class="div1">

                <span>HELLO</span>

                    <div class="div2">

                            <span>HOLA</span>

                    </div>

            </div>

   </div>

**CSS**\*{

    margin: 0;

    padding:0;

}

.div-container{

    border:1px solid red;

    width:70%;

    font-size:2em;

}

.div1{

    border:1px solid green;

    width:50%;

}

.div2{

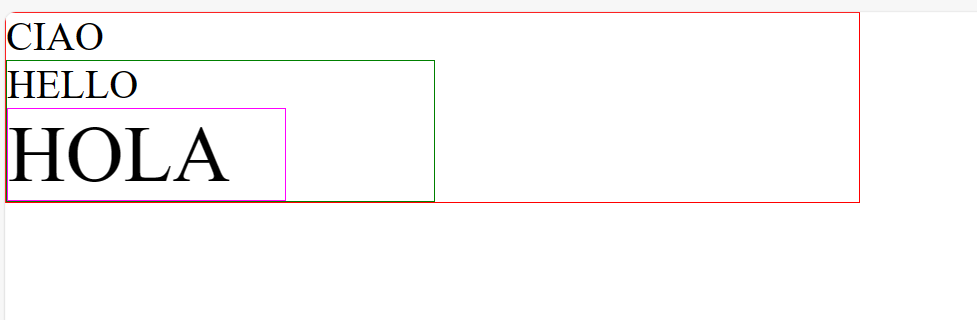
    border:1px solid fuchsia;

    width:65%;

    font-size:2em;

}

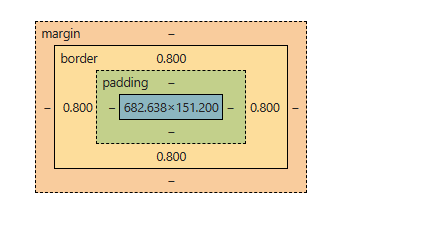
**BROWSER**

****

**La width del div più esterno .div-container è pari al 70% della width del body.**

**In altre parole:**

**70% di 975.2px è pari a 682.64**

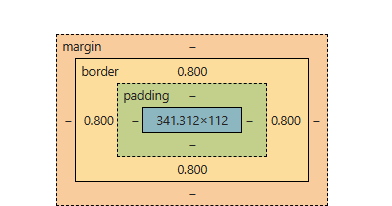
****

**Perfect!**

**Ora consideriamo la width del div innestato, ossia di div .div1.**

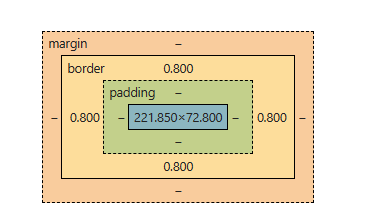
**Tale div ha width pari al 50% del div contenitore, ossia di div .div-container.**

**La width del div .div1 sarà quindi pari al 50% di 682.64px = 341.32px**

****

**Perfect!**

**Infine, vediamo che la width del div .div2, innestato al div .div1, sarà pari al 65% della width del div .div1, ossia 65% di 341.312px = 221.86px.**

****

**Perfect!!**

**Quindi, utilizzando le percentuali, avremo che il metro di paragone è sempre il div container.**

**E se volessimo calcolare la width del div innestato .div2 rispetto alla width del body invece che rispetto alla width del div contenitore .div1?**

**FACILE! Si utilizza l’unità di misura vw.**

**CSS**

\*{

    margin: 0;

    padding:0;

}

.div-container{

    border:1px solid red;

    width:70%;

    font-size:2em;

}

.div1{

    border:1px solid green;

    width:50%;

}

.div2{

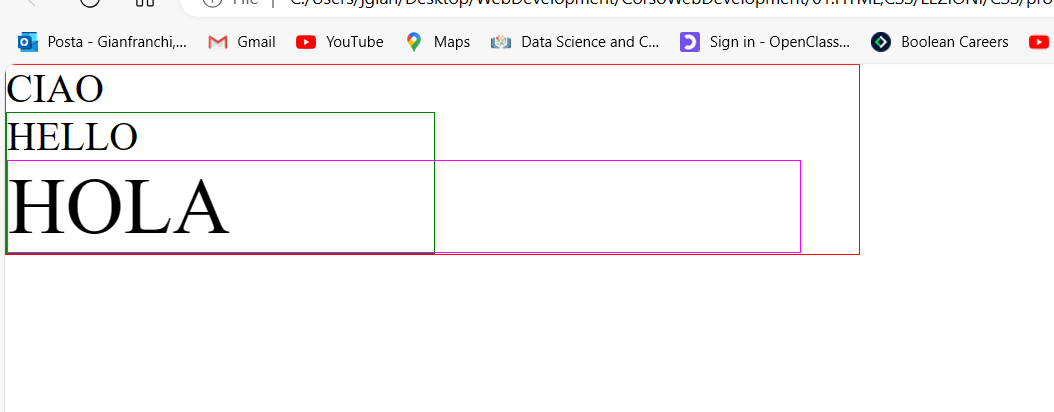
    border:1px solid fuchsia;

    width:65vw;

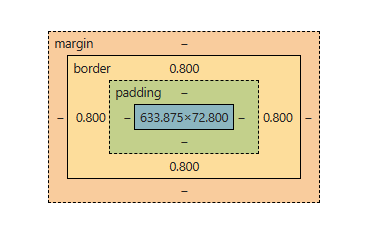
    font-size:2em;

}

**BROWSER**

****

**Dato che il nostro body ha width pari a 975.2px, avremo che la larghezza del div .div2 sarà pari al 65% di 975.2px, ossia a 633.88px.**

****

**Perfect!!**